

## TD et TP 3 PLSQL

Soit le schéma de base de données suivant :

Client : (**id**, prenom, nom, email, ville, );

Commande: (**id**, client\_id, date\_achat, reference);

commande\_ligne: (**id**, commande\_id, nom\_produit, quantite, prix\_total);

Produit : (**nom\_produit**, prix\_unitaire, quantite\_stock, en\_promotion)

1. Ajouter les colonnes quantite\_stock et en\_promotion dans la table Produit. Initialiser la colonne en\_promotion avec la valeur 0.
2. SQL dynamique : on décide de mettre certains produits en promotion. La condition pour qu'un produit soit mis en promotion dépend de plusieurs paramètres (le stock, la période de l'année...)  
Ecrire la procédure **sp\_DetermineProduitEnPromo**(p\_whereclause, p\_promo out) qui permet de déterminer si un produit est en promotion ou pas. Elle met à jour la colonne en\_promotion=1 si la condition p\_whereclause est vérifiée ; =0 si non.  
p\_promo contient le nombre de produits en promotion. Son contenu sera affiché.
3. Gestion des stocks :
  - Créer une séquence seq\_CommandeLigne
  - Modifier la procédure sp\_InsertCommandeLigne du TP1 qui permet d'insérer une ligne de commande :
    - utiliser la séquence seq\_CommandeLigne pour générer automatique p\_id. La procédure devient alors sp\_InsertCommandeLigne (p\_commande\_id, p\_nom\_produit, p\_quantite)
    - Si une ligne de commande existe déjà (nom\_produit déjà commandé pour la commande commande\_id), mettez à jour la quantité de la ligne de commande ; la ligne ne sera alors pas insérée. Si le nom\_produit n'existe pas pour cette commande, insérée la ligne.
  - Créer un trigger trg\_GestionStock qui va permettre de gérer les stocks : avant qu'une nouvelle ligne de commande soit ajoutée, vérifiez que la quantité commandée <= quantite\_stock du produit. Si ok, la ligne de commande est ajoutée et la quantité est mise à jour la table produit dans le cas contraire, une exception est levée.
4. Ecrire une procédure sp\_DateDernierAchat qui affiche la liste de tous les produits et la date du dernier achat. La procédure devra retourner 9999-01-01 si le produit n'a jamais été commandé.
5. Sql dynamique : on souhaite cloner certaines tables de la base de données. Ecrire la procédure sp\_clonetable(nom\_table) qui prend en paramètre le nom d'une table et crée son clone nom\_table\_clone. Utiliser la vue all\_tab\_columns.
6. Modifier la procédure sp\_clonetable(nom\_table) pour intégrer les clés primaires et clés étrangères. Tester la procédure avec les tables client, commande et commande\_ligne.
7. Créer un package avec tous les objets créés en TP